```
import java.util.*;
public class GameOfLife
{
	Scanner sc=new Scanner(System.in);

	int [][] board;
	int row,col,k=1;

	public void setData()  //taking initial input
	{
		System.out.print("\n\t\t\t Enter the number of rows : ");
		int row=sc.nextInt();

		System.out.print("\n\t\t\t Enter the number of columns : ");
		int col=sc.nextInt();


		int[][] board=new int[row][col];

		System.out.print("\n\t\t\t Press 1 for LIVE \n\t\t\t Press 2 for DEAD \n\n\t\t\t Enter
the STATE of cell : " );

		for(int i=0;i<row;i++)
			for(int j=0;j<col;j++)
				board[i][j]=sc.nextInt();

		this.row=row;
		this.col=col;
		this.board=board;

	}

	public void getData()  //printing the board of cells
	{
		for(int i=0;i<row;i++)
		{
			System.out.println("\t\t\t\t\t");
			for(int j=0;j<col;j++)
			{
				if(j==0)
					System.out.print("\t\t\t\t\t"+board[i][j]+"\t");
				else
					System.out.print(board[i][j]+"\t");
			}
```

```
            }
            System.out.println();
        }

        public void print()
        {
            if(k<row)
                System.out.println("\n\n\t\t\t State : "+(k++));

            getData();
        }

        public void stateOfCell()
        {
            int counts=0;
            System.out.println("\n\t\t\t Enter the location of cell  :  ");
            System.out.println("\n\t\t\t Enter Row: ");
            int ro=sc.nextInt();
            System.out.println("\n\t\t\t Enter Column : ");
            int co=sc.nextInt();

            if(ro-1<row && co-1<col)
            {
                for(int i=0;i<row;i++)
                {
                    for(int j=0;j<col;j++)
                    {
                        if(board[ro-1][co-1]==0)
                            counts=0;
                        else
                            counts=1;
                    }
                }

                if(counts==0)
                    System.out.print("\n\t\t\t*** Given cell is DEAD ***\n");
                if(counts==1)
                    System.out.print("\n\t\t\t*** Given cell is LIVE ****\n");
        System.out.print("\n\t\t\t_____________________________________________\n");
            }
            else
            {
                System.out.println("\n\t\t\t Incorrect Location. \n\t\t\t Please, enter valid
row and column : ");
```

```
                    stateOfCell();
                }
        }

        public void getState()
  {
                int exit;

                do
                {
                    System.out.println("\n\t\t\t_____ GAME OF LIFE
_____\n");

                    System.out.println("\n\t\t\t 1.Next State \n\t\t\t 2.Check State of a cell ");
                    System.out.print("\n\t\t\t Enter your choice : ");

                    int ch=sc.nextInt();

                    switch(ch)
                    {
                        case 1:
                            gameOfLife();
                            print();
                            break;

                        case 2:
                            gameOfLife();
                            stateOfCell();
                            break;

                        default :
                            System.out.print("\n\t\t\t Invalid input !!! ");
                            break;
                    }

                    System.out.print("\n\t\t\t Press 1 to continue \n\t\t\t Press 0 to Exit  : ");
                    exit=sc.nextInt();

     }while(exit!=0);

  }


        public void  gameOfLife()
```

```
        {
                for (int i=0;i<row;i++)
                {
                        for (int j=0;j<col;j++)
                        {
                                int box=board[i][j];
                                transition(i,j,box);
                        }
                }
        }

        public void transition(int i,int j,int box)
        {
                int count=0;

                int a = i - 1;
                int b = i + 1;
                int c = j - 1;
                int d = j + 1;

                if (a >= 0 && board[a][j]==1)
                        count++;

                if(b <row && board[b][j]==1)
        count++;

       if(c >=0 && board[i][c]==1)
          count++;

       if(d <col && board[i][d]==1)
           count++;

//lower right side diagonal

        if(j>=0 && j<col-1 && i>=0 && i<row-1)
        {
           if(board[b][d]==1)
                    count++;

        }

//upper left side diagonal

        if(i>0 && i<row && j>0 && j<col)
```

```
        {
           if(board[a][c]==1)
                    count++;

        }

//lower left side diagonal

        if(j>0 && j<col && i>=0 && i<row-1)
        {
           if(board[b][c]==1)
                    count++;

        }

//upper side right diagonal

        if(i>0 && i<row && j>=0 && j<col-1)
        {
           if(board[a][d]==1)
                    count++;
        }

        if(box==1)
        {
           if(count<2)
                    board[i][j]=0;

           if(count>3)
                    board[i][j]=0;

           if(count==2|| count==3)
                    board[i][j]=board[i][j];

        }

        if(box==0)
        {
           if(count==3)
                    board[i][j]=1;
        }

          }
```

```
        public static void main(String[] args)
        {
                Scanner sc=new Scanner(System.in);

                GameOfLife obj=new GameOfLife();

                obj.setData();
                obj.getState();

        }

}
```

o/p